

**Basics of XML technology –
XML in education, with special emphasis on programming
education**

Menyhárt, László Gábor

Theses of PhD Dissertation

Eötvös Loránd University

Department of Media & Educational Informatics



Supervisor: Dr. Remzső Tibor

PhD School of Informatics

Dr. Csuhaj Varjú Erzsébet

PhD program of Basics and methodology of informatics

Dr. Horváth Zoltán

2018

Introduction

Extensible Markup Language, briefly XML [HL1, HL2] can be found in many areas of information technology. This open source language was published on 10 February 1998 at first, and version 1.0 has become W3C Recommendation on 26 November 2008. XML can describe structured data in text format, so it appeared in a lot of scope. It turns up from communication of standard web-services through configuration files to end user file formats, so it can be found in computers of the least experienced users, too. I started to deal with the „Latin” of the IT world, because XML, like Latin, is a flexible and practical language and quickly spreaded. There is a widespread experience in all IT fields. It is present in the development of all major information technology companies, in web communication, in solving integration problems and in structured descriptions of configuration tasks, but it does not get the attention in IT education as much as some picked up novelty – sometimes just for short time –.

History and motivations

The basic theme of my university thesis writing in 2003 as an informatics teacher was XML. I created a curriculum about XML that was stored in native XML database and I implemented a web-application to search the content with XML-based technologies.

The topic was so appealing that I chose this for my topic in doctoral school, too. I started my research with title „Basics of XML technology” as parsing text files and XML compression capabilities.

During my teaching work at current Department of Media and Educational Informatics, I started to use XML more and more frequently and to publish my methodological results. Now it is time to bring together my results in education, research and development in this dissertation.

Life-like examples in education methodology were always close to me and this subject with XML was more engaging to deal for me. Deeper understanding of technology is possible with implementation of own function libraries. Because XML documents are texts and string processing is a topic in programmer training, it is easy to understand that XML can be a practical example.

XML can help us at development of teaching methods. The technology-supported teaching of the Programming Fundamentals at Faculty of Informatics on the University can be a distancing step away from instinctive programming.

My first XML based, teaching material content management application made in the mentioned university thesis is no longer available, but I use the curriculum since then in my yearly course. I am also attending at other courses, so I am trying to incorporate the XML in other classes. From the educational and methodological point of view, new ideas have been formulated that can be useful.

Objective

In my dissertation, I summarize my efforts in teaching XML, using XML in education and other methodological innovations. The main topic of the dissertation is based on XML and the technologies around it. I present several educational tool I developed and tested with my students. I am writing about the application methodology of these software.

Theses

1. Thesis about XML education

Basics of XML technologies could be presented at the early stages of the current universtiy information technology training with my curriculum on XML and related technologies, so the students could build to this during their later studies.

I did a research to gather information about XML teaching materials in educational organizations. The collected data of my case study are listed in the appendix of the dissertation, and I also expounded the knowledge in text I acquired when I was looking at the form in which the XML was taught.

The method of data collection was an online research, so the information here were published on Internet and I found them. I used Google basically; I started with free texts, then thematically on universities and filtering their pages. These starting points are available in the list of links. I continued with searching in their databases if it was available; or I have read the available descriptions and subject-disciplines.

XML is available many times in a lot of educational materials of the universities. Somewhere it comes up only on a few lessons, elsewhere it is a full course. Somewhere it is mentioned as a data presentation option, but it can also be compared to structured databases.

With several of my colleagues, we have noticed that mentioning XML is unavoidable in today's IT education, so it comes up again and again, it is compared with and built on it. But I found that XML is not available in mandatory courses and only a few educate all XML technology.

I have been teaching a special course with my own curriculum since 2002/03 Spring, its name is „Data management – XML” and it has two codes in ELTE's NEPTUN IKI-AXFG and TANM-INF-300-XML.

Feedbacks from students show that they love this course and they found it interesting and useful.

2. Thesis about using XML to teach string processing

XML provides a good example to training text processing and project work.

String processing can be found in several subjects. During their practical training, a lot of times the task is only a simple problem to solve that is not enough spectacular. Instead of the experience, they focus on the algorithmic solution of the problem and they try to solve strained problems or simple, abstract tasks. The readCount algorithm is an example from the recommended note [HP1] to process text sequentially.

I figured out a new problem to solve in this topic, in which XML must be read and HTML should be created. I worked out the lesson plan, the curriculum and the presented and distributed framework. I tried to teach it on a course and I conducted a survey among the participating students. I published this topic in my article [SP1].

So there is an example that teaches text processing with XML. This is a good example based on the feedbacks of the students. In addition, the second group working on project provided better evaluations when completing the questionnaire.

3. Thesis about algorithm markup language

Education of beginner programmers can be helped by Algorithm Markup Language (AML), which is an extension of XML with defining its own types.

Teaching programming is not only about algorithmic thinking, but also about source code making, compiling, building, running, making document, testing and debugging. Different schools have chosen other and other language to teach as the „first language”. Earlier Pascal was the most dominant language, but at most places it was replaced some C-like language. In the meantime, there were published graphical interfaces that try to make the algorithms more spectacular and playful, so students could understand them easier. There are some applications where students can define commands with text and graphical objects appear on the surface, as in Logo. But there are some where commands are also graphical objects which can be build with drag&drop method. For example: Scratch, Google Blockly, Snap! or App Inventor 2. Children can create games with this to computers and mobile tools.

At the beginning of the educational process the structure of structured programming languages can be presented with the new markup language that I defined and presented in my article [SP2]. Parts of each control elements can be understood. Syntax can be easily verified, and even at editing the offered commands can be used. Source codes and other files can be generated. Because the XML file contains many extra tag names in addition to the really useful data, we should write a lot, but fortunately there are many XML editors that support fast syntax writing with code completion at editing time. It also helps the teaching because this can be work in a students’ native language, as opposed to source code that correlates to English. XML editors can also display the documentation in annotations that help the understanding. This language would be an unnecessary step for advanced programmer students to whom testing and debugging are more important, which are not supported in AML, because unidirectional code generation is possible only. This tool should be used as a game as calculating sticks or egg cups are used in first class of the primary school at teaching tens and ones. It can be good in a few introductory lessons. For example it could be used in the first few lessons on the Programming Fundamentals course not to horrify the students with the C source code syntax, but take the first steps with code completion to understand the structure of the code. Since HTML editing is already being studied in a lot of places – because it is graduation topic – structure of a Markup Languages is known, to which they can bind the newly presented control commands and their structure. Thus, AML syntax has known

structure – even if it is not simpler – and editors code completion capabilities help to select the usable option with displaying short descriptions in mother tongue (or in English).

4. Thesis about my algorithm visualisation application

The effectiveness of learning can be enhanced by my XML-based application for visualising and editing algorithms.

The presented AML is an XML from which different outputs can be created with XSL transformation, so an appropriate text file must be edited. In some cases, it would be better to have a graphical application to build the algorithms. My article [SP3] is about this web-application.

At the beginning of structured programming education, it is impossible to miss deepening algorithmic thinking and approach. We have several options to represent algorithms [HP2], including flowchart [HP3, HL3], Jackson Diagram [HP4, HP5], pseudo-code [HP6], and structure known as Nassi-Shneiderman Diagram [HP3]. Graphic appearance may not be enough, it would be much better if editing happened in graphical view as well.

At first I was working with Scalable Vector Graphics (SVG) format, then I continued with HTML to further expand it with JavaScript. I have prepared a prototype of an AML code editor, which starts from visualized NSD and updates it after every editing step.

Nassi-Schneiderman Diagrams (NSD) can be used to visualize functions and procedures without having to learn a language [HP7, HP8, HP9]. There are a lot of tools which facilitate representing algorithms in NSD format, such as Structorizer [HL4]. Several of similar tools have been presented as a BsC thesis or student research in Scientific Students' Associations Conference, see [HL5, HL6]. I am interested in platform-independent web solutions, so I've created one of these to work on any of today's web browsers. I have chosen a server-client platform as basis of the development, the algorithm is stored on the server side in the previously presented AML. XSL transformation is here as well, so NSD will appear as HTML on client side, more accurately it presented as `div-s` and `table-s`.

The XML-based AML gives the opportunity to generate a source code, but I just focused on displaying and modifying algorithms. You can use this tool to edit the algorithm stored in the AML format so that you do not need to know this new language. Users start editing

instructions with clicks. A graphically displayed algorithm is easier to understand and the editing option also helps the learning process.

There are some tools that can be used to create a Nassi-Shneiderman diagram. Some of them can draw only a complete algorithm. Others can be used for editing. Paint or any other graphic program can be used, just like tables in Microsoft Word or any other office application. Microsoft Visio [HL7] supports this with a simple template. While SmartDraw [HL8] or Structorizer [HL4] applications are also good for editing an algorithm. This way, the middle of the algorithm can be modified at any time. Lot of times these applications must be downloaded and installed. Nowadays, more people have started developing web applications for this area.

The tool I created is a platform-independent client-server solution that allows users to display and edit algorithms in a browser without installation any application. You can copy the NSD image from the screen into the documentation. This can be a good choice for students to easily generate NSDs. It can also be useful for teachers who want to display algorithms with a simple tool. Of course, I've extended the web page with features such as user management and saving image.

The beginning of the thesis has been published by others, for example, Gábor Törley in his dissertation [HP10], while the self-developed editor application was shown that it can display and edit the structure of algorithms in NSD format.

5. Thesis about my application to support the programming process

My XML-based application built to the analogy of the tools used by the methodologies and spreaded in industry supports teaching of systematic programming.

Students should solve IT tasks on lesson "Programming Fundamentals", which usually get a text description, from which they prepare the specification, algorithm and implement the code. I began to think about how I can support students' thinking and help to record the tasks to be solved, possibly to speed up implementation and documentation creation. To solve this problem, I began to deal this parallel with the previous algorithm visualising and editing program. Later I build together the two ideas.

Teaching programming to IT students at the University must cover the entire development process, but in the process of systematic programming we must focus on algorithmization. [HP11] I found that most of my students are only interested in the essence of coding, which comes from the text of the task and tries to avoid specification creation and algorithm design. It was my idea that students would pay more attention to specification and algorithm if we were to help solve their later assignments. The essence of the support consists of code generation from the algorithm and preparation of the main elements of the documentation. First, the information need to be collected in an Excel spreadsheet, from which documentation can be generated without presenting the program, ie screenshots. First my article [SP4] reported this in more detail.

In the second step, I've presented a more complex web-based application in the article [SP5]. I made my students try to use the application when they learnt only about syntax of the programming language. There was a group where education was in English, because students came from different countries (England, Brazil, Iran, Cameroon, Macedonia, Germany, Saudi Arabia, Turkey, USA, Vietnam ...). Here it was especially good that the algorithm was written with no pseudonym code because we did not know each other's mother tongue, and English could have caused confusion because it is very similar to the programming language in the source code. [HP12]

We do not have to teach the tools of programming (programming language and its environment), but programming itself. We must develop a mindset, a style of programming in the audience and must focus on algorithmization in the process of systematic programming. [HP11] It is important that the student understands the algorithm and can code it in some language. They need to understand basic algorithms (algorithm patterns) and use them as building blocks to make more complex algorithms. Therefore, these algorithm patterns should be analyzed, checked and examined. During education, it is important to start creating an application only after developing algorithmic skills, but it is necessary for practical testing to develop the programming skills of students simultaneously.

I introduced my ideas, the Excel-based support package as a line leader. Then I built this on a web platform using the algorithm – as Nassi-Shneidermann diagram – editor application that was mentioned in the previous thesis. I answered the questions that have arisen, I wrote about my experiences and the feedbacks. Based on the positive welcome and feedbacks of the PFW, I can say that it helps in teaching systematic programming.

Publications of the author

- [SP1] Menyhárt L.: Teaching string processing by using XML in: Nagib Callaos , Hsing-Wei Chu , Belkis Sánchez , Michael Savoie , Andrés Tremante (editor) Proceedings of the 7th International Multi-Conference on Engineering and Technological Innovation . Conference place, date: Orlando, USA, 2014.07.15 -2014.07.18. Orlando: International Institute of Informatics and Systemics (IIS), 2014. pp. 73-76., ISBN:978-1-941763-07-0
- [SP2] Menyhárt L.: Can a language be before "the first programming language"? in: TEACHING MATHEMATICS AND COMPUTER SCIENCE 9:(2) pp. 209-224., 2011
- [SP3] Menyhárt L.: Nassi-Schneiderman diagram in HTML based on AML In: ACTA DIDACTICA NAPOCENSIA 6:(3) pp. 19-26., 2013, ISSN: 2065-1430
- [SP4] Menyhárt L., Pap Gné.: How can we get our students to think while we help their work too?: Document based development In: Nagib Callaos , José Vincente Carrasquero , Ángel Oropeza , Andrés Tremante , Fridrich Welsch (editor) Proceedings of the 7th International Multi-Conference on Society, Cybernetics and Informatics . Conference place, date: Orlando, USA, 2013.07.09 -2013.07.12. Florida: International Institute of Informatics and Systemics (IIS), 2013. pp. 97-102., ISBN:ISBN-13:978-1-936338-83-2
- [SP5] Menyhárt L., Pap Gné.: Presentation of improved version of guide application for teaching programming fundamentals In: Nagib Callaos , Hsing-Wei Chu , Belkis Sánchez , Michael Savoie , Andrés Tremante (editor) Proceedings of the 7th International Multi-Conference on Engineering and Technological Innovation . Conference place, date: Orlando, USA, 2014.07.15 -2014.07.18. Orlando: International Institute of Informatics and Systemics (IIS), 2014. pp. 77-82., ISBN:978-1-941763-07-0

References

Publications

- [HP1] Pap, Gné., Szlávi, P., Zsakó, L.: Módszeres programozás – Szövegfeldolgozás, 14. Mikrológia, 2000
- [HP2] Furman, B. J.: Notes on Algorithms, Pseudocode, and Flowcharts, 2010, http://www.engr.sjsu.edu/bjfurman/courses/ME30/ME30pdf/Notes_on_Algorithms.pdf

- [HP3] Nassi, I., Shneiderman, B.: Flowchart techniques for structured programming, ACM SIGPLAN Notices, Volume 8 Issue 8, 12–26, 1973
- [HP4] Jackson, M.A.: Principles of Program Design, Academic Press, 1976
- [HP5] Jackson, M.: A system development method In Cambridge University Press, 1982, http://www.ferg.org/papers/jackson--a_system_development_method.pdf
- [HP6] Cormen, T. H., Leiserson, C. E., Rivest R. L.: Algoritmusok, Műszaki Könyvkiadó, Budapest, 1997, p 2-4.
- [HP7] Hendrix, T.D., Cross, J.H.: "Language Independent Generation of Graphical Representations of Source Code", Proceeding CSC '95 Proceedings of the 1995 ACM 23rd annual conference on Computer science Pages 66-72, 1995
- [HP8] Csepregi, Sz., Dezső, A., Gregorics, T., Sike, S.: Automatic Implementation of Service Required by Components, Workshop on Property Verification for Software Components and Services, PROVECS, <http://lina.atlanstic.net/provecs/2007/provecs2007proceedings.pdf>, 2007
- [HP9] Averbukh, V., Bakhterev, M.: The analysis of visual parallel programming languages, ACSIJ Advances In: Computer Science: an International Journal, Vol.2, Issue3, No. 4, 2013, ISSN : 2322-5157
- [HP10] Törley, G.: Vizualizáció a Programozástanításban, doctoral thesis, 2014, <http://www.doktori.hu/index.php?menuid=193&lang=HU&vid=12890>
- [HP11] Szlávi P., Zsakó L.: Módszeres programozás: Programozási bevezető, 18. Mikrológia, 1996
- [HP12] Weise, M.: A Model for Teaching Informatics to German Secondary School Students In: English-language Bilingual Education, Proceedings of the 6th International Conference ISSEP 2013; Oldenburg, Germany, February 26–March 2, 2013/Diethelm et al. (Eds.)/ Potsdam: Universitätsverlag Potsdam, 2013/ S.127-137

Links

- [HL1] W3C - Extensible Markup Language, <http://www.w3.org/XML/>
- [HL2] XML - W3C Recommendation, <http://www.w3.org/TR/xml/>
- [HL3] Flowchart, <https://en.wikipedia.org/wiki/Flowchart>
- [HL4] Fisch, B.: Structorizer, <http://structorizer.fisch.lu/>, 2009-
- [HL5] Molnár, T.: StukiMania, <http://stukimania.hu>, 2012-
- [HL6] Németh B.: DAVIK, (Dinamikus Algoritmus Vizualizációs Keretrendszer) Dynamic Algorithm VISualisation Framework, OTDK:

http://otdk2013.gdf.hu/sites/default/files/OTDK_2013_Informatika_Tudomanyi_Szekcio_GDF_Programfuzet.pdf, szakdolgozat:

http://www.inf.elte.hu/karunkrol/szervezet/dekanihivatal/to/diplomamunkak/diplomamunkak/Lapok/11122_diplomak.aspx

[HL7] Microsoft Visio Template for Nassi-Schneiderman Diagram,
<http://users.tm.net/tonietienne/>

[HL8] SmartDraw, <http://www.smartdraw.com/resources/tutorials/nassi-shneiderman-diagrams/>